

DIGITAL SIGNAL PROCESSOR

TECHNICAL FIELD OF THE INVENTION

- 5 This invention relates to a digital signal processor, and in particular to multiplier circuitry within a digital signal processor.

BACKGROUND OF THE INVENTION

- 10 High performance digital signal processing applications can conveniently be implemented using programmable logic devices. Therefore, Stratix™ programmable logic devices from Altera Corporation include DSP blocks, in the form of high-performance embedded DSP units, which are optimized for applications such as rake receivers, orthogonal frequency division multiplexing transceivers, and image
15 processing applications.

One defining feature of a DSP block is the bit length of the words which it handles. For example, a 16 bit DSP architecture stores data in the form of 16 bit words, and allows easy manipulation of such 16 bit words.

- 20 However, although a 16 bit architecture is sufficient for many applications, and therefore is in common use, there are a significant number of applications for which a 16 bit architecture is insufficient. For example, when using a digital signal processor to perform inversion of a matrix, the use of a 16 bit architecture may be insufficient to
25 calculate the coefficients of the resulting matrix with the required accuracy.

- In such circumstances, a floating point DSP processor can be used to obtain the result to the required accuracy, but such processors are expensive and inconvenient. Alternatively, a 16 bit architecture can be used to perform the required operations, but
30 this is a slow process. To illustrate this, two multiplicands, each of up to 32 bits, can each be divided into two 16 bit words. The two words forming the first multiplicand must then be multiplied in turn by the two words forming the second multiplicand, so that four multiplication operations are required. The result of multiplying the most significant bits of the two multiplicands must then be shifted 32 bit positions to the left,
35 while the two results of multiplying the most significant bit from one multiplicand with the least significant bits from the other multiplicand must be added together and shifted

16 bits to the left. Finally, these intermediate results must be added together to form the final result. This means that, if a 16 bit multiplication occupies one clock cycle of the digital signal processor, a 32 bit multiplication occupies nine clock cycles or more, depending on the data moving and shifting capabilities of the digital signal processor.

5

SUMMARY OF THE INVENTION

According to the present invention, there is provided a digital signal processor architecture, and a method of operation of such an architecture, which allows the digital
10 signal processor to be used efficiently for multiplying words which are longer than the word length for which the architecture is primarily designed.

According to an aspect of the invention, the multiplication unit has a register file which is adapted to store data words of a first length, and a multiplier which is adapted to
15 multiply together data words of a second length, the second length being twice the first length. In a first mode, the architecture multiplies data words of the first length, by extending them to the second length. In a second mode, the architecture multiplies data words of the second length, by retrieving each of the data words in two parts, each part being of the first length.

20

The digital signal processor architecture of the present invention may form part of an embedded digital signal processor block in a programmable logic device, or may form part of a dedicated digital signal processor integrated circuit.

BRIEF DESCRIPTION OF DRAWINGS

25

Figure 1 is a block schematic diagram of a multiplication unit in a digital signal processing block in accordance with the present invention.

30 Figure 2 is a flow chart showing a method of operation of the multiplication unit of Figure 1.

Figure 3 is a block schematic diagram of a multiplication unit in a digital signal processing block in accordance with an alternative embodiment of the present
35 invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

Figure 1 is a block schematic diagram showing a multiplication unit in a digital signal processor according to the present invention. Data, for use in the multiplication unit 10, can be stored in registers within register files 12, 14. For ease of illustration, these two register files 12, 14 are shown separately, and in the following description it will be assumed that the first multiplicand is stored in the first register file 12, while the second multiplicand is stored in the second register file 14. In practice, in one preferred embodiment of the invention, there is a single dual ported register file, in which both multiplicands are stored, and which allows two registers to be read out at the same time.

The registers of the first register file 12 are connected through a first multiplexer 16, whose output is supplied to a first multiplication register 18. The registers of the first register file 12 are also connected to a second multiplexer 20. The output of the second multiplexer 20, and the most significant bit of the output of the first multiplexer 16, are connected to a third multiplexer 22, and the output of the third multiplexer 22 is supplied to a second multiplication register 24.

Similarly, the registers of the second register file 14 are connected to a fourth multiplexer 26, whose output is supplied to a third multiplication register 28. The registers of the second register file 14 are also connected to a fifth multiplexer 30, and the output of the fifth multiplexer 30 and the most significant bit of the output of the fourth multiplexer 26 are connected to a sixth multiplexer 32, whose output is supplied to a fourth multiplication register 34.

The multiplexers 16, 20, 22, 26, 30, 32 all operate under the control of a control unit 36, as will be described in more detail below.

The contents of the first multiplication register 18, second multiplication register 24, third multiplication register 28 and fourth multiplication register 34 are supplied to a multiplier block 38, the outputs of which are supplied to two 32 or more bit accumulation units 40, 42.

Figure 1 shows a 16 bit architecture. That is, each of the registers in the register files 12, 14 is 16 bits wide, while the various datapaths and the multiplication registers 18,

24, 28, 34 are each also 16 bits wide. However, the multiplier 38 is a 32-bit multiplier, that is, it is able to multiply two 32 bit numbers, and the accumulators 40, 42 are 32 bits wide. Although the invention is described based upon a 16 bit architecture, the same principle can be applied to other architectures. For example, an 18 bit DSP
 5 architecture can be used to multiply together two 36 bit numbers, or a 32 bit architecture can be used to multiply together two 64 bit numbers.

Figure 2 is a flow chart showing a method of operation of the multiplication unit 10.

10 Initially, in step 50, it is determined whether the unit is operating in a 32 bit multiplication mode. If not, that is, if the unit is operating in a 16 bit multiplication mode, the process passes to step 52.

In step 52, the data is loaded into the multiplication registers. More specifically, the first
 15 multiplicand is loaded from a register in the first register file 12, and the first multiplexer 16 is controlled by the control unit 36 such that this data word is loaded into the first multiplication register 18. In 16 bit mode, there is no need to retrieve a second data word from the register file 12, and so the third multiplexer 22 is controlled such that the most significant bit of the first multiplicand is passed through from the first multiplexer
 20 16. This bit value is then stored in each of the 16 bit positions in the second multiplication register 24. Thus, the first and second multiplication registers 18, 24 together contain a sign extended version of the first multiplicand.

At the same time, the second multiplicand is loaded from the intended register in the
 25 second register file 14 through the fourth multiplexer 26 into the third multiplication register 28. As before, the sixth multiplexer 32 is controlled so that the most significant bit of the data loaded into the third multiplication register 28 is also loaded into each bit position in the fourth multiplication register 34.

30 As described above, sign extended versions of the first and second 16 bit multiplicands are stored, with the second multiplication register 24 being filled with the most significant bit of the first multiplicand, and the fourth multiplication register 34 being filled with the most significant bit of the second multiplicand. This will allow the multiplier to produce a signed result of the 16 bit multiplication.

35

In other embodiments of the invention, the second multiplication register 24 and the fourth multiplication register 34 could each be filled with zeroes. In this case, the multiplication will produce an unsigned result.

5 Alternatively, the multiplier 38 could be modified so that, in this 16 bit multiplication mode, it can produce a signed result from the product of the first and second multiplicands, irrespective of the contents of the second multiplication register 24 and the fourth multiplication register 34.

10 The principle is that, when the contents of the first and second multiplication registers 18, 24 are multiplied by the contents of the third and fourth multiplication registers 28, 34, it remains easily possible to obtain an output which is the product of the first and second multiplicands.

15 In a further alternative embodiment of the invention, the first 16 bit multiplicand could be stored in the second multiplication register 24, and the second 16 bit multiplicand could be stored in the fourth multiplication register 34. In this case, the first multiplication register 18 and the third multiplication register 28 should be filled with zeroes, again so that, when the contents of the first and second multiplication registers
20 18, 24 are multiplied by the contents of the third and fourth multiplication registers 28, 34, it remains easily possible to obtain an output which is the product of the first and second multiplicands.

In this case, unlike the other cases discussed above, the desired multiplication result
25 will appear as the upper 32 bits of the multiplier output.

In step 54 of the process, the data stored in the first and second multiplication registers 18, 24 is multiplied by the data stored in the third and fourth multiplication registers 28, 34, in the 32 bit multiplication unit 38.

30

In step 56 of the process, the multiplier 38 outputs the multiplication result, under the control of the control unit 36. In the preferred embodiment of the invention, in which the first and second multiplicands are stored in the first and third multiplication registers respectively, it is only the 32 bit result obtained from multiplying the contents of the first
35 multiplication register 18 and the third multiplication register 28 which has any

significance. This can be supplied to one or both of the accumulation units 40, 42, depending upon a control signal from the control unit 36.

5 However, in the alternative embodiment described above, in which the first and second multiplicands are stored in the second and fourth multiplication registers respectively, the desired multiplication result will appear as the upper 32 bits of the multiplier output, and so these bits must be routed to one or both of the accumulation units 40, 42.

10 If instead it is determined in step 50 that the device is operating in a 32 bit multiplication mode, then an additional step is required to load the data to be multiplied.

Thus, in step 58, the first part of the data to be multiplied is loaded. In step 58, the 16 bit data word forming the most significant bits of the first 32 bit multiplicand is loaded from the respective register of the first register file 12, through the second multiplexer 15 20, and the third multiplexer 22 is controlled so that this data word is stored in the second multiplication register 24. At the same time, the data word forming the 16 most significant bits of the second multiplicand is loaded from a register in the second register file 14, through the fifth multiplexer 30 and the sixth multiplexer 32 is controlled so that this data word is stored in the fourth multiplication register 34.

20 This process occupies one clock cycle and, in a second clock cycle in step 60, the data words representing the least significant bits of the two multiplicands are loaded. Thus, the data word forming the least significant bits of the first multiplicand is loaded from a register in the register file 12 through the first multiplexer 16 into the first multiplication 25 register 18. At the same time, the 16 bit word forming the least significant bits of the second multiplicand is loaded from a register in the second register file 14 through the fourth multiplexer 26 into the third multiplication register 28.

30 In one preferred embodiment of the invention, the two 16 bit data words forming the 32 bit first multiplicand will usually be stored in adjacent registers in the register file 12, while the two 16 bit data words forming the 32 bit second multiplicand will usually be stored in adjacent registers in the register file 14, although this is not necessarily the case.

35 Then, as before, in step 62, the contents of the first and second multiplication register 18, 24 are multiplied by the contents of the third and fourth multiplication registers 28,

34 in the multiplier 38. Again, as before, under the control of the control unit 36, the multiplication result is supplied to the accumulation units 40, 42. In this case, the 64 bit multiplication result may be divided between the accumulated units 40, 42, with the 32 most significant bits in one accumulation unit and the 32 least significant bits in the other accumulation unit. Alternatively, in some situations it may be sufficient to output only the 32 most significant bits, or the 32 least significant bits, to one accumulation unit 40, 42 or the other.

Although the invention has been described herein with reference to one preferred embodiment, it will be appreciated that other implementations are also possible.

For example, Figure 3 shows an alternative embodiment of the invention, in which the first and third multiplication registers 18, 28 are removed, and the multiplexers 22, 32 are replaced by multiplexers 72, 74 positioned on the outputs of the second and fourth multiplication registers 24, 34 respectively.

In this alternative embodiment of the invention, in the 16 bit multiplication mode, the first and second multiplicands are loaded directly into the 16 least significant bits of the two respective 32 bit inputs of the multiplier 38, while the multiplexers 72, 74 are controlled so that the most significant bits of these two multiplicands are repeated in the respective 16 most significant bits of the two multiplier inputs.

In the 32 bit multiplication mode, the 16 bit words forming the most significant bits of the two multiplicands are loaded into the multiplication registers 24, 34. Then, in the next cycle, the 16 bit words forming the least significant bits of the two multiplicands are loaded directly into the multiplier 38, while the multiplexers 72, 74 are controlled so that the 16 bit words forming the most significant bits of the two 32 bit multiplicands are also loaded into the multiplier 38 from the registers 24, 34.

The multiplier 38 can then multiply the two 32 bit multiplicands.

The invention therefore provides an architecture which allows a multiplication to be carried out efficiently, even when the multiplicands are up to twice the length of the stored data words.